基于分解和多策略变异的多目标差分进化算法 *

童旅杨 [a]，董明刚 [a, b†]，敬　超 [a, b]

(桂林理工大学 a. 信息科学与工程学院; b. 广西嵌入式技术与智能系统重点实验室，广西 桂林 541004)

摘　要：差分进化是一种有效的优化技术，已成功用于多目标优化问题。但也存在 Pareto 最优集合的收敛慢和多样性差等问题。针对上述不足，提出了一种基于分解和多策略变异的多目标差分进化算法(MODE/DMSM)。该算法利用基于分解的方法将多目标优化问题分解为多个单目标优化问题；通过高效的非支配排序方法选择具有良好收敛性和多样性的解来指导差分进化过程；采用了多策略变异方法来平衡进化过程中收敛性和多样性。在 ZDT 和 DTLZ 的 10 个测试函数上的仿真结果表明，本文算法在 Parato 最优集合的收敛性和多样性优于其他六种代表性多目标优化算法。

关键词：多目标优化；差分进化；分解；多策略变异

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2018.01.0028

Multiobjective differential evolution algorithm based on decomposition and multi-strategy mutation

Tong Lyuyang[a], Dong Minggang[a, b†], Jing Chao[a, b]

(*a. College of Information Science & Engineering*, *b. Guangxi Key Laboratory of Embedded Technology & Intelligent System*, *Guilin University of Technology*, *Guilin Guangxi* 541004, *China*)

Abstract: Differential evolution algorithm is an efficient optimization technique that has been successfully applied to multiobjective optimization problems. However, there are also some defects, i. e. the slow convergence and poor diversity of the Pareto optimal set. Addressing these issues, this paper presented a multiobjective differential evolution algorithm based on decomposition and multi-strategy mutation (MODE/DMSM) . MODE/DMSM utilized the decomposition-based approach to decompose a multiobjective optimization problem into multiple single-objective optimization problems. Moreover, MODE/DMSM adopted the efficient non-dominated sorting approach to select solutions which have both good convergence and diversity to guide the differential evolutionary process. Eventually, MODE/DMSM employed the multi-strategy mutation approach to balance the convergence and diversity in the evolutionary process. The results of simulations on 10 test functions of ZDT and DTLZ show that MODE/DMSM outperforms than the other six representative multiobjective optimization algorithms in terms of the good convergence and diversity of the Pareto optimal set.

Key words: multiobjective optimization; differential evolution; decomposition; multi-strategy mutation

0　引言

在优化问题中存在同时需要优化两个或者多个互相冲突的目标问题，这类问题称为多目标优化问题[1](multiobjective optimization problem, MOP)。多目标问题没有单个最优解，其最优解是一个 Pareto 解集。Pareto 解集的收敛性和多样性是衡量多目标算法的重要指标。

多目标进化算法(multiobjective evolutionary algorithms, MOEA)是基于种群模拟生物进化全局优化算法，已经广泛应用解决多目标问题。当前多目标进化算法主要分为三类：

a）基于 Pareto 占优的多目标进化算法采用 Pareto 支配来选择非支配解，如 NSGAII[2]和 SPEA2[3]。

b)基于评估指标的多目标进化算法采用某种评估指标来选择非支配解，如 IBEA[4]和 HypE[5]。

c)基于分解技术的多目标进化算法将多目标问题分解成一组单目标优化问题。如 Zhang 和 Li 提出了一种基于分解的多目标进化(MOEA/D)算法[6]，该算法将多目标问题分解成多个单目标问题来求解，然后用基因遗传算法(genetic algorithm，GA)

收稿日期：2018-01-17；**修回日期：**2018-03-09　**基金项目：**国家自然科学基金资助项目（61563012，61203109）；广西自然科学基金资助项目（2014GXNSFAA118371，2015GXNSFBA139260）；广西嵌入式技术与智能系统重点实验室基金资助项目

作者简介：童旅杨（1992-），男，湖北监利人，硕士研究生，主要研究方向为进化计算、多目标优化、机器学习；董明刚（1977-），男（通信作者），教授，博士，主要研究方向为进化计算、机器学习、并行和分布式计算（d2015mg@qq.com）；敬超（1983-），男，讲师，博士，主要研究方向为大数据&云计算、节能计算和优化调度.

求解这些单目标问题。

差分进化算法[7](differential evolution，DE) 是一种简单高效的全局优化进化算法（evolutionary algorithm， EA），已经应用于多目标问题求解。Zhong 和 Zhang 提出了一种带有随机编码策略的自适应多目标差分进化算法(AS-MODE)[8]，使用 DE 中一种简单的变异策略来生成新种群。Venske 提出一种自适应的多目标差分进化算法(ADEMO/D)[9]。Jiang 提出了一种基于两阶段策略和小生境指导方案的改进MOEA/D(MOEA/D-TPN)[10]，该算法中 DE 作为基础优化算法。文献[11]提出了基于分解的差分进化算法(MOEADDE)[11]，使用 DE 代替 MOEA/D 中 GA 算法，通过分解的方法更新种群。总体来看，多目标进化算法主要采用单一的变异策略，导致 Pareto 解集的收敛性和多样性不够理想。

为了提高 Pareto 解集的收敛性和多样性，本文在 MOEADDE 的基础上提出了一种基于分解和多策略变异的多目标差分进化算法(MODE/DMSM)。算法采用改进 Tchebycheff 分解方法将一个多目标问题分解为一组单目标优化子问题，同时生成一系列均匀分布的权重向量。为了兼顾收敛性和多样性，算法采用高效非支配排序对种群进行排序并保存了非支配解和一部分较好的解。算法利用多策略变异差分进化算法来引导种群向真实的 Pareto 前沿收敛同时均匀分布，实现收敛性和多样性的平衡。与其他典型的多目标优化算法进行仿真比较，结果表明本文提出的算法在解决多目标问题时在收敛性和多样性具有一定的优势。

1　相关背景

1.1　多目标优化问题

一般的多目标问题，以最小化形式为例，具有 n 个决策变量和 M 个优化目标，可表示为如下形式：

$$\begin{aligned}&\text{minimize } \ \mathrm{F}(x)=(f_1(x),\dots,f_m(x))\\&\text{subject to：} \ x\in\Omega\end{aligned} \tag{1}$$

其中：$x=\{x_1,\dots,x_n\}\in\Omega\subset R^n$ 称为决策向量；$F(x)=(f_1(x),\dots f_m(x))\in R^m$ 称为目标向量；Ω 为决策空间。以下是多目标问题引出的几个相关定义和术语。

定义 1　Pareto 支配[12]。假设 $x,x^*\in\Omega$ 是多目标问题的可行解，并且 x^* 对 x 是 Pareto 支配的，记为 $x^*\prec x$，当且仅当

$$\forall i=1,\dots,m,\ f(x^*)\le f(x)\wedge F(x^*)\ne F(x) \tag{2}$$

定义 2　Pareto 最优解集[12]。Pareto 最优解集（Pareto Set, 简称 PS）是所有 Pareto 最优解的集合。

$$PS=\{x^*\mid\neg\exists x\in\Omega,x\prec x^*\} \tag{3}$$

定义 3　Pareto 最优前沿[12]。Pareto 最优前沿(Pareto front, PF)是 Pareto 最优解集 PS 中的解对应目标函数值组成的集合。

$$PF=\{F(x^*)\mid x^*\in PS\} \tag{4}$$

1.2　基于 Tchebycheff 多目标分解方法

分解方法将多目标优化问题转换成单目标优化问题，给每个个体分配一个不同的权重向量来缓解目标之间的冲突。常见的分解方法有切比雪夫(Tchebycheff)方法[6]。传统的 Tchebycheff 分解方法公式如下：

$$\begin{aligned}&\text{minimize}:\mathrm{g}(\mathrm{x}\mid\lambda,z^*)=\max_{1\le i\le m}\{\lambda_i\left|f_i(x)-z_i^*\right|\}\\&\text{subject to：} \ x\in\Omega\end{aligned} \tag{5}$$

其中：λ_i 为第 i 个权重向量；$z^*=(z_1^*,\dots z_m^*)$ 为更新过程中的理想点；$z_i=\min\{f_i(x)\mid x\in\Omega\}$。

1.3　差分进化算法

差分进化算法是 Storn 和 Price 提出的一种基于种群的随机搜索算法，主要通过变异、交叉和选择等操作来进行智能搜索。DE 与其他进化算法的主要区别在于变异操作，目前被广泛采用的变异操作如下：

DE/rand/1[13]:

$$v_{i,G}=x_{r_1,G}+F\cdot(x_{r_2,G}-x_{r_3,G}) \tag{6}$$

DE/rand-to-best/1[13]:

$$v_{i,G}=x_{r_1,G}+F\cdot(x_{best,G}-x_{r_1,G})+F(x_{r_2,G}-x_{r_3,G}) \tag{7}$$

DE/best/1[13]:

$$v_{i,G}=x_{best,G}+F\cdot(x_{r_1,G}-x_{r_2,G}) \tag{8}$$

其中：r_1，r_2 和 r_3 是从种群规模 NP 中随机产生的 3 个互不相等的整数，G 为当前种群的进化代数，F 为缩放因子，$best$ 表示当前最好个体的下标。v 和 x 分别表示变异向量和种群的个体。

2　基于分解和多策略变异的多目标差分进化算法

2.1　多目标分解方法

Tchebycheff 函数是在基于分解的 MOEAs 中使用的最常见类型的聚合函数之一。 在本文中采用了文献[14]中改进的 Tchebycheff 函数[14]，这种改进的 Tchebycheff 函数在原始函数上有两个优点。 首先，均匀分布的权向量导致目标空间中搜索方向的均匀分布；其次，每个权重向量对应于位于 PF 上唯一的解[15]。这两个优点在一定程度上缓解了多样性保存的困难。改进的 Tchebycheff 公式如下：

$$\begin{aligned}&\text{minimize}:\mathrm{g}(\mathrm{x}\mid\lambda,z^*)=\max_{1\le i\le m}\{\left|f_i(x)-z_i^*\right|/\lambda_i\}\\&\text{subject to：} \ x\in\Omega\end{aligned} \tag{9}$$

其中：z^* 为理想点，λ_i 为权重向量。$\lambda_i\ge0$，如果 $\lambda_i=0$，将设置 $\lambda_i=10^{-6}$。Das 和 Dennis[16]提出标准边界交叉（NBI）技术被广泛的应用在目标空间生成权重向量。

$$\begin{cases}\lambda_i=(\lambda_i^1,\dots,\lambda_i^M)\\ \lambda_i^j\in\{\dfrac{0}{H},\dots,\dfrac{H}{H}\},\sum_{j=1}^{M}\lambda_i^j=1\end{cases} \tag{10}$$

其中：将每维目标分割成 H 个相等的区间，M 为目标个数，则生成的权重向量的数量 $N=C_{H+M-1}^{H}$。假设将三维空间每维目标分割成 2 个相等的区间，生成权重向量的个数 $N=C_{2+3-1}^{2}$=6，图 1 是在三维空间上生成权重向量的示意图。

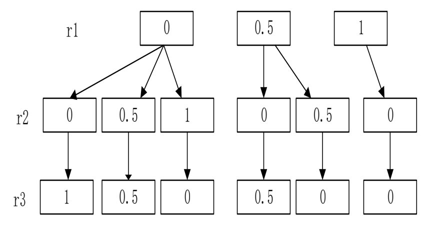


图 1 权重向量生成示意图

本文采用了上述的方法生成权重向量，首先，均匀分布的权重向量使得在目标空间中搜索方向是均匀分布。其次，每个权重向量和理想点让每个个体趋向 Pareto 前沿。在一定程度上，增加了种群的多样性和收敛性

2.2 高效非支配排序策略

本文运用了高效非支配排序[17]来保存比较优的解。这种方法首先根据第一个目标值对种群 *P* 中的 *N* 个解进行升序排序，其中 *N* 是种群大小。如果两个解的第一个目标函数值相同，则按第二个目标函数值排序。一直持续到种群中的所有个体都被排序。如果解在所有目标函数值中都相同，它们的顺序可以是任意的。对于这个经过排序的种群 *P*，如果 $m<n$，解 p_m 将永远不会被解 p_n 支配，因为 p_m 中至少存在一个目标，其值小于 p_n 中相同目标。表明这两种解之间只存在两种可能的关系：p_m 支配 p_n，或者 p_m 和 p_n 互不支配。本文通过高效的非支配排序通过对种群进行排序，选出前面 *N/2* 个个体作为 *NDbest* 解集，既保留了非支配解集，又保留了一些较好的解来增加 *NDbest* 解集的多样性。高效非支配排序的伪代码如下：

高效非支配排序策略（ENS）

```
F=empty; /*F 为前沿的集合*/
把种群 P 按照第 1 个目标函数值进行升序排序，如果两个解的第一个目标函数值相同，则按第二个目标函数值排序，一直持续到种群中的所有个体都被排序；
for all  P[n] ∈ sorted P do
    x=size(F);
    k=1;
    while true do
        从 F[k] 的最后一个到第一个解与比较 P[n]；
        if  F[k] 中没有解支配 P[n]  then
            把 P[n] 添加到 F[k]；
            break;
        else
            k++;
            if k>x then
                x+1 ;
                把 P[n] 添加到新的前沿;
                break;
            end if
        end if
    end while
end for
return F;
```

另外，本文的差分进化算法需要从 *NDbest* 解集中随机选择个体来引导进化过程，在 *NDbest* 解集中的非支配解集来引导种群收敛，又同时考虑了 *NDbest* 解集中一些较好的解来增加种群的多样性。*NDbest* 解集将会用于下面的多策略变异操作。

2.3 多策略变异操作

差分进化算法中变异策略会影响整个种群的更新，因此选择合适的变异策略是改善整个算法性能的关键。“DE/rand/1”变异策略针对种群的多样性有很好的效果，但是收敛性不足可能不能推进收敛过程，而如果选择收敛性强的策略“DE/ rand -to-best/1”，可能又因为多样性差从而导致种群早熟收敛。因此，在选择合适的变异策略上，尽可能兼顾其收敛性和多样性，从而有效的推进进化的过程。

本文提出了一种多策略变异。在进化初期，让 DE/rand/1 变异策略发挥其全局搜索的能力增加种群的多样性，随着进化过程的进行，本文提出了一种新的变异策略“DE/rand-to-*NDbest*/1”，通过上一代高效非支配排序选择出 *NDbest* 解集来指导进化过程进一步增强收敛性，从而达到了收敛性和多样性的平衡，最后运用二项式交叉和多项式变异。“DE/rand-to-*NDbest*/1”变异策略如下：

$$v_{i,G}=x_{r_1,G}+rand\cdot(x_{NDbest,G}-x_{r_1,G})+F(x_{r_2,G}-x_{r_3,G}) \quad (11)$$

其中：*NDbest* 表示当前种群高效非支配排序最优集中随机个体的下标，rand 表示 0 到 1 的随机数。二项式交叉如下：

$$\bar{y}_{i,G}^{j}=\begin{cases}v_{i,G}^{j} & if\,(rand_j(0,1)\le CR)or(j=j_{rand}),\ j=1,2,...D\\ x_{i,G}^{j} & otherwise\end{cases} \quad (12)$$

其中 *CR* 是交叉概率，j_{rand} 为 1 到 *D* 中随机产生的一个整数，以保证尝试向量 $\bar{y}$ 与当前个体 *x* 不相同。多项式变异公式如下：

$$y_{i,G}^{j}=\begin{cases}\bar{y}_{i,G}^{j}+\sigma_i^j(b_i^j-a_i^j) & if\,(rand_j(0,1)\le p_m)\\ \bar{y}_{i,G}^{j} & otherwise\end{cases} \quad (13)$$

$$\sigma_i^j=\begin{cases}(2+rand)^{\frac{1}{\eta+1}}-1 & if\,(rand\le 0.5)\\ 1-(2-2*rand)^{\frac{1}{\eta+1}} & otherwise\end{cases} \quad (14)$$

其中 p_m 为变异比率，b_i 和 a_i 分别是第 *i* 个决策变量的上下界，η 是分布指数。多策略变异的伪代码如下：

多策略变异操作

```
if  rand ≤ 1-(G/G_max)^2 /*rand 是 0 到 1 随机生成的随机数*/
    v_{i,G} = x_{r1,G} + F·(x_{r2,G} - x_{r3,G}) /*DE/rand/1*/
else
    v_{i,G} = x_{r1,G} + rand·(x_{NDbest,G} - x_{r1,G}) + F(x_{r2,G} - x_{r3,G})
    /*DE/rand-to-NDbest/1*/
end if
根据式（12）进行二项式交叉；
最后根据式（13）（14）进行多项式变异。
```

2.4 MODE/DMSM 算法

MODE/DMSM 算法总体流程如下:a)初始化种群；b)采用 2.2 中方法对种群进行高效的非支配排序选出前 *N/2* 个个体作为 *NDbest* 集；c)采用 2.3 中方法多策略变异产生子代 *y*；d)采用 2.3 中方法改进的 Tchebycheff 分解方法更新种群中的个体；e)采用 2.2 中方法对种群进行高效的非支配排序选出前 *N/2* 个个体作为 *NDbest* 集；f)如果满足停止条件，算法停止，输出 *PF*，否则返回步骤 c)继续执行。MODE/DMSM 算法的伪代码如下：

Algorithm: pseudo code of MODE/DMSM

随机初始化种群 *P*，设置种群数量 *N* 和最大评估次数 *MAX_FES*,计算群体每个个体适应度值和每个目标最优适应度值 $z^*=(z_1^*,...,z_m^*)$，随机生成 *N* 个权重向量 $\lambda=(\lambda_1,...,\lambda_N)$。计算权重向量之间的欧式距离，并且找出每个权重向量距离最近的 *T* 个权重向量 $\lambda^{i_1},...,\lambda^{i_T}$，其索引是 $B(i)=\{i_1,...,i_T\}$。设置邻域个体选择概率 δ 为 0.9，子代替换父代的最大数量 n_r 为 2 ，*FES=N*。

通过 2.2 中方法高效的非支配排序（*ENS*）进行非支配排序选出 *N/2* 个个体作为 *NDbest*。

```
while FES<Max_FES do
    for i = 1 to N do
        随机产生 0 到 1 的随机数 rand 生成
        A = { B(i)    if (rand < δ)
            { {1,...N}  otherwise
        按 2.3 中方法多策略变异 DE 更新种群生成新的 解 y。
        如果 y 超出了边界的范围 Ω，在边界里随机生成新值代替 y。
        计算个体的适应度值，更新 z* = (z1*,...,zm*) 。
        for j = 1 to m do
            if  zj* > fj(y) then
                zj* = fj(y)
            end if
        end for
        更新种群的解,设置 c=0,
        while  ¬(c == nr | isempty(A)) do
            随机从 A 中取出索引 k;
            if    g(y | λk, z*) ≤ g(xk | λk, z*)
                xk = y ;
                c=c+1;
            end if
            从 A 中清除索引 k;
        end while
        FES=FES+N;
    end for
    通过 2.2 中方法高效的非支配排序（ENS）进行非支配排序选出 N/2
    个个体作为 NDbest
end while
```

2.5 时间复杂度分析

假设算法的种群大小为 *N*，目标数为 *M*，领域大小为 T。根据文献[18]可知，MOEADDE 的时间复杂度为 $O(MNT)$ [18]。由于 *T*<*N* 且 *T* 与 *N* 相关，MOEADDE 的时间复杂度为 $O(MN^2)$。MOEADDE 在种群进化中始终采用单一的变异算子，而 MODE/DMSM 在 MOEADDE 基础上增加了多策略变异算子没有额外增加算法时间复杂度；另外，MODE/DMSM 运用了高效非支配排序，根据文献[17]可知，高效非支配排序所需要的最好情况的时间复杂度是 $O(MN\sqrt{N})$，最坏情况的时间复杂度是 $O(MN^2)$ [17]，故 MODE/DMSM 的时间复杂度为 $O(MN^2)$。

3 仿真实验及结果分析

为了测试本文算法 MODE/DMSM 的性能，选取 4 种代表性多目标算法进行对比，分别是 MOEAD[6]、NSGAIII[19]、RVEA[20]、MOEADDE[11]、NSLS[21]和 MOEAIGDNS[22]。所有的算法都是在多目标进化算法开源平台 PlatEMO[23]上运行。实验仿真的环境都是在 Intel(R) Core(TM)i7-7700HQ CPU、16G 内存、2.8GHz 主频，win10 64 位操作系统的计算机上 Matlab R2016a 运行。

3.1 测试函数及评价指标

本文采用文献[24]提出 ZDT1-ZDT4、ZDT6[24]和文献[25]提出 DTLZ1-DTLZ5[25]的 10 个测试函数来测试本文算法 MODE/DMSM 的有效性。评价指标采用世代距离 GD(Generation Distance)[26]、反向世代距离 IGD(Inverted Generation Distance)[27]和超体积 HV(Hypervolume)[28]来评价解集的收敛性和多样性。其计算公式如下：

$$GD(P,P^*)=\frac{\sum_{v\in P}d(v,P^*)}{|P|} \tag{15}$$

$$IGD(P^*,P)=\frac{\sum_{v\in P^*}d(v,P)}{|P^*|} \tag{16}$$

$$HV(P)=Vol\left(\bigcup_{x\in P}[f_1(x),z_1^*]\times...\times[f_m(x),z_m^*]\right) \tag{17}$$

其中:*P* 为所得的 Pareto 前沿，P^* 为最优的 Pareto 前沿，$d(v,P^*)$ 表示目标空间上 *v* 与 P^* 中最近解之间的欧式距离，$Vol(\cdot)$ 记为 Lebesgue 测度。

3.2 实验参数设置

算法在每个测试函数上独立运行 30 次，设置种群规模 *NP* 为 100，最大函数评估次数 *Max_FES* 为 30000，多项式变异 η=20、$p_m=1/D$。文本算法 MODE/DMSM 的缩放因子 *F* 为 0.5，交叉概率 *CR* 为 0.2；MOEADDE 的缩放因子 *F* 为 0.5，交叉概率 *CR* 为 1；本文 MODE/DMSM 和 MOEADDE 中的邻域个体选择概率 δ 为 0.9，子代替换父代的最大数量 n_r 为 2；MOEAD、NSGAIII、RVEA、NSLS 和 MOEAIGDNS 中交叉率 $p_m=1$、$\eta_c=30$；RVEA 中惩罚因子变化率为 $\alpha=2$，参考点调整频率 $f_r=0.1$。

3.3 实验结果分析

MODE/DMSM 与 MOEAD、NSGAIII、RVEA、MOEADDE、NSLS、MOEAIGDNS 测试于 ZDT 和 DTLZ 测试函数上分别在

GD、IGD 和 HV 标准下的 30 次独立运行的平均值和标准差统计结果如表 1-3 所示，本文采用 Wilcoxon 秩和检验进行比较突出算法之间差异的显著性，显著性水平为 0.05，表中“+”、“-”、“≈”分别表示其他算法明显优于、明显劣于和近似于 MODE/DMSM，其中最好的结果用不同颜色的字体表示。

从表 1 中的数据在 GD 标准下可以看出，本文提出的算法 MODE/DMSM 分别在 2 个测试函数(ZDT6、DTLZ5)上近似于 MOEAD；在 2 个测试函数(ZDT3、ZDT6)上近似于 NSGAIII；在 1 个测试函数(ZDT6)上近似于 RVEA；在 1 个测试函数(ZDT6)上近似于 MOEADDE，在两个测试函数(ZDT3、ZDT6)上近似于 MOEAIGDNS。在其余的测试函数下，MODE/DMSM 优于 MOEAD、NSGAIII、RVEA、MOEADDE、NSLS、MOEAIGDNS。总体上，MODE/DMSM 在 8 个测试函数(ZDT1~4、DTLZ1~3、DTLZ5)上得到最好的结果。

从表 2 中的数据在 IGD 标准下可以看出，MODE/DMSM 在 8 个测试函数（ZDT1~4、ZDT6、DTLZ1、DTLZ 3、DTLZ5）上优于 MOEAD；MODE/DMSM 在 5 个测试函数（ZDT1、ZDT 3~4、DTLZ3~4）上优于 NSGAIII；MODE/DMSM 在 7 个测试函数（ZDT1~4、ZDT6、DTLZ3、DTLZ 5）上优于 RVEA；MODE/DMSM 在 8 个测试函数（ZDT1~4、DTLZ1~4）上优于 MOEADDE；MODE/DMSM 在 8 个测试函数（ZDT1~4、ZDT6、DTLZ1、DTLZ3~4）上优于 NSLS, MODE/DMSM 在 6 个测试函数（ZDT1~2、ZDT4、ZDT6、DTLZ3~4）上优于 MOEAIGDNS。总体上，MODE/DMSM 在 5 个测试函数(ZDT1~4、DTLZ3)上得到最好的结果。

从表 3 中的数据在 HV 标准下可以看出，MODE/DMSM 在 8 个测试函数（ZDT1~4、ZDT6、DTLZ1、DTLZ 3、DTLZ5）上优于 MOEAD；MODE/DMSM 在 7 个测试函数（ZDT1~4、DTLZ1、DTLZ3、DTLZ4）上优于 NSGAIII；MODE/DMSM 在 8 个测试函数(ZDT1~4、ZDT6、DTLZ1、DTLZ3、DTLZ5)上优于 RVEA；MODE/DMSM 在 8 个测试函数（ZDT1~4、DTLZ1~4）上优于 MOEADDE；MODE/DMSM 在 9 个测试函数（ZDT1~4、ZDT6、DTLZ1~4）上优于 NSLS 和 MOEAIGDNS。总体上，MODE/DMSM 在 6 个测试函数(ZDT1~4、DTLZ1、DTLZ3)上得到最好的结果。

表 1 测试函数在 GD 标准下实验统计结果

Problem	M	MOEAD	NSGAIII	RVEA	MOEADDE	NSLS	MOEAIGDNS	MODE/DMSM
ZDT1	2	4.1652e-4 (1.19e-4) -	9.2005e-5 (3.45e-5) -	3.2302e-3 (8.28e-4) -	1.0803e-3 (6.49e-4) -	5.1423e+0 (9.68e-1) -	7.3084e-5 (2.47e-5) -	1.2125e-5 (9.29e-6)
ZDT2	2	9.4598e-4 (6.22e-4) -	6.3058e-5 (2.31e-5) -	3.3958e-3 (1.44e-3) -	9.5758e-4 (5.12e-4) -	6.8252e+0 (1.37e+0) -	4.6044e-5 (2.01e-5) -	7.8375e-6 (5.78e-6)
ZDT3	2	1.5232e-3 (1.94e-3) -	6.6748e-5 (5.54e-5) ≈	3.3952e-3 (1.25e-3) -	2.1661e-3 (2.21e-3) -	6.0136e+0 (1.26e+0) -	5.0231e-5 (1.33e-5) ≈	4.9466e-5 (1.14e-5)
ZDT4	2	1.7281e-3 (1.04e-3) -	2.9247e-4 (1.72e-4) -	2.1164e-3 (8.00e-4) -	2.8039e-2 (1.71e-2) -	3.6696e-1 (9.17e-2) -	3.7083e-4 (2.27e-4) -	4.4888e-5 (1.59e-5)
ZDT6	2	8.1940e-4 (2.22e-4) ≈	6.2187e-5 (6.02e-5) ≈	3.8055e-3 (1.37e-3) ≈	2.4356e-3 (1.09e-2) ≈	7.5549e-2 (6.84e-2) -	2.0133e-4 (2.96e-4) ≈	1.6643e-2 (2.24e-2)
DTLZ1	3	3.3229e-4 (1.46e-4) -	2.6684e-4 (1.14e-4) -	2.8930e-4 (1.34e-4) -	2.8448e-2 (1.28e-1) -	5.1296e-2 (3.14e-2) -	4.3197e-4 (1.95e-4) -	1.8284e-4 (8.07e-6)
DTLZ2	3	5.0524e-4 (2.64e-6) -	5.0326e-4 (5.75e-6) -	5.1379e-4 (2.22e-5) -	8.1839e-4 (6.15e-5) -	1.2919e-3 (1.92e-4) -	6.8633e-4 (4.44e-5) -	4.9107e-4 (1.71e-5)
DTLZ3	3	1.7300e-1 (3.98e-1) -	3.3643e-1 (8.69e-1) -	1.8290e-1 (1.84e-1) -	4.4169e-1 (8.35e-1) -	5.6068e+0 (1.01e+0) -	1.8665e-1 (5.67e-1) -	2.6823e-2 (6.84e-2)
DTLZ4	3	3.7487e-4 (1.81e-4) ≈	4.9923e-4 (6.50e-5) -	5.1017e-4 (6.98e-6) -	8.6356e-4 (4.21e-4) -	5.2988e-3 (4.00e-3) -	6.2166e-4 (1.62e-4) -	4.7772e-4 (1.59e-5)
DTLZ5	3	1.6000e-5 (3.16e-5) -	2.5470e-4 (1.15e-4) -	4.2376e-2 (1.54e-2) -	8.8713e-5 (2.49e-5) -	2.6775e-4 (5.77e-5) -	3.7241e-5 (1.11e-5) -	6.5693e-6 (1.58e-6)
+/-/≈		0/8/2	0/8/2	0/9/1	0/9/1	0/10/0	0/8/2	

表 2 测试函数在 IGD 标准下实验统计结果

Problem	M	MOEAD	NSGAIII	RVEA	MOEADDE	NSLS	MOEAIGDNS	MODE/DMSM

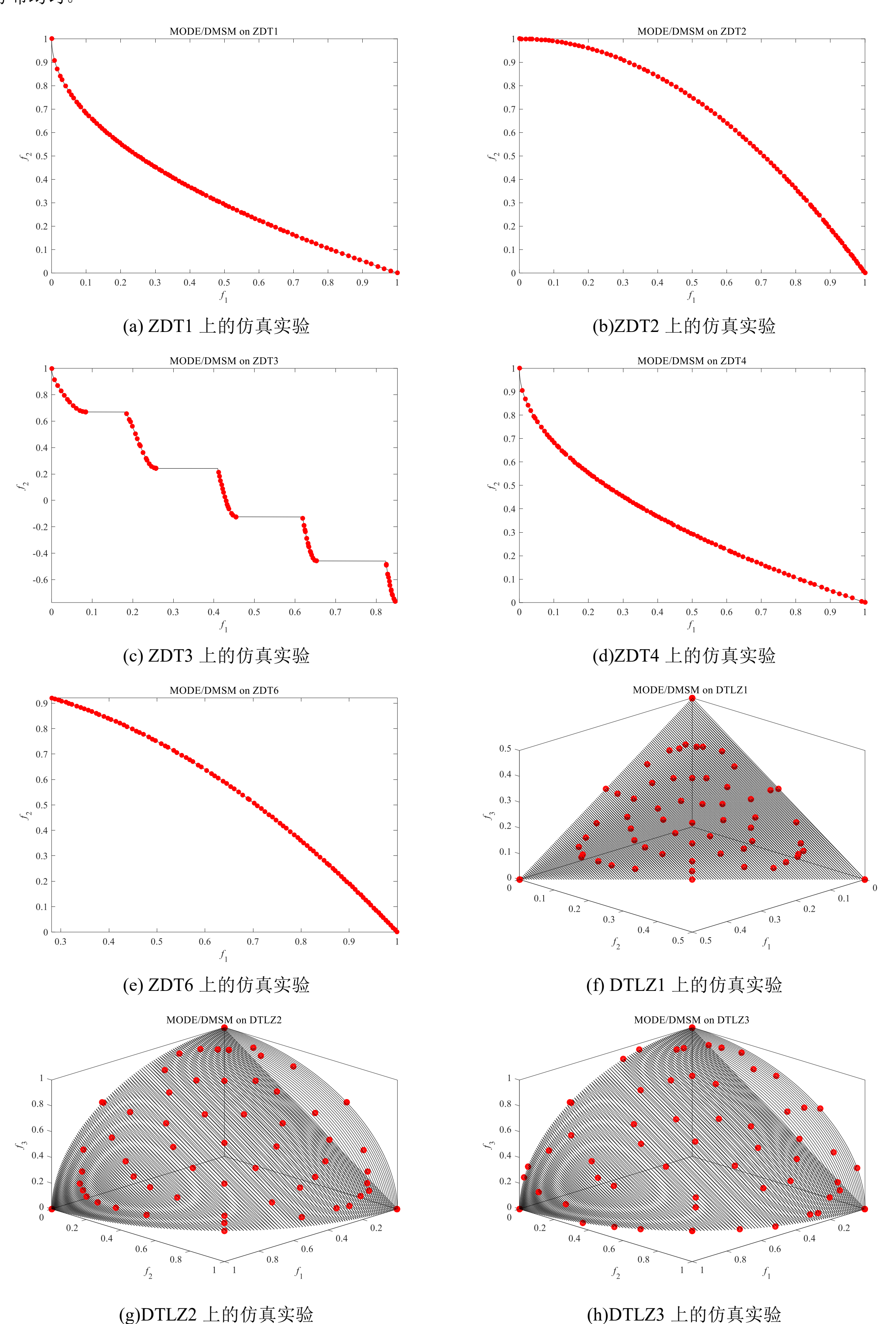
ZDT1	2	1.6602e-2 (2.12e-2) -	5.3384e-3 (5.40e-3) -	3.2468e-2 (2.01e-2) -	1.1745e-2 (5.73e-3) -	7.1899e+0 (2.06e+0) -	6.4159e-3 (8.65e-3) -	3.9653e-3 (3.96e-5)
ZDT2	2	4.8749e-2 (7.15e-2) -	4.9106e-3 (3.15e-3) ≈	4.4291e-2 (5.38e-2) -	9.6554e-3 (3.74e-3) -	8.5175e+0 (1.79e+0) -	1.6885e-2 (2.58e-2) -	3.9602e-3 (4.39e-5)
ZDT3	2	2.8498e-2 (2.75e-2) -	6.0005e-2 (7.59e-2) -	8.2320e-2 (5.14e-2) -	2.8014e-2 (2.04e-2) -	7.6289e+0 (1.93e+0) -	3.6516e-2 (3.76e-2) ≈	1.1575e-2 (3.71e-4)
ZDT4	2	2.1374e-2 (1.26e-2) -	1.0630e-2 (2.06e-2) -	4.6528e-2 (4.93e-2) -	2.1136e-1 (1.14e-1) -	8.3338e-1 (1.97e-1) -	9.0156e-3 (1.43e-2) -	4.1580e-3 (1.29e-4)
ZDT6	2	7.3477e-3 (1.46e-3) -	3.1696e-3 (2.56e-4) +	2.7225e-2 (7.75e-3) -	3.1064e-3 (9.11e-6) +	6.5237e-3 (1.77e-3) -	3.5963e-3 (3.88e-4) -	3.2963e-3 (9.80e-5)
DTLZ1	3	2.0988e-2 (4.36e-4) -	2.0825e-2 (3.44e-4) ≈	2.1209e-2 (2.23e-3) ≈	4.9475e-2 (6.55e-2) -	2.8327e-1 (1.96e-1) -	2.0138e-2 (7.85e-4) +	2.0750e-2 (7.37e-5)
DTLZ2	3	5.4467e-2 (1.43e-6) +	5.4497e-2 (9.15e-5) +	5.4508e-2 (1.25e-4) +	7.5988e-2 (8.30e-4) -	5.6350e-2 (7.54e-4) ≈	5.1570e-2 (5.61e-4) +	5.6600e-2 (7.36e-4)
DTLZ3	3	1.2828e+0 (2.71e+0) -	5.6226e-1 (8.11e-1) -	8.7055e-1 (9.05e-1) -	2.2512e+0 (4.18e+0) -	1.7435e+1 (3.85e+0) -	2.7297e-1 (4.12e-1) -	2.5802e-1 (5.05e-1)
DTLZ4	3	3.8164e-1 (3.41e-1) ≈	1.6816e-1 (2.10e-1) -	5.4487e-2 (4.54e-5) +	1.5330e-1 (8.37e-2) -	2.9700e-1 (6.82e-2) -	1.6585e-1 (2.10e-1) -	6.0017e-2 (1.63e-3)
DTLZ5	3	3.3778e-2 (6.92e-5) -	1.3344e-2 (1.68e-3) +	7.9796e-2 (1.26e-2) -	1.4375e-2 (9.07e-5) +	5.1314e-3 (2.40e-4) +	4.1794e-3 (5.24e-5) +	2.2831e-2 (8.29e-5)
+/-/≈		1/8/1	3/5/2	2/7/1	2/8/0	1/8/1	3/6/1	

表 3　测试函数在 HV 标准下实验统计结果

Problem	M	MOEAD	NSGAIII	RVEA	MOEADDE	NSLS	MOEAIGDNS	MODE/ DMSM
ZDT1	2	8.5700e-1 (1.60e-2) -	8.6904e-1 (4.76e-3) -	8.2655e-1 (1.59e-2) -	8.5580e-1 (9.74e-3) -	0.0000e+0 (0.00e+0) -	8.6834e-1 (7.21e-3) -	8.7143e-1 (9.01e-5)
ZDT2	2	4.7374e-1 (7.90e-2) -	5.3484e-1 (7.60e-3) -	4.7327e-1 (5.53e-2) -	5.2297e-1 (8.25e-3) -	0.0000e+0 (0.00e+0) -	5.1430e-1 (3.73e-2) -	5.3829e-1 (5.78e-5)
ZDT3	2	9.8476e-1 (3.82e-2) -	9.5422e-1 (9.67e-2) -	9.0636e-1 (5.99e-2) -	9.7045e-1 (3.85e-2) -	0.0000e+0 (0.00e+0) -	9.8196e-1 (4.61e-2) -	1.0201e+0 (1.83e-4)
ZDT4	2	8.4079e-1 (1.60e-2) -	8.6296e-1 (1.58e-2) -	8.2191e-1 (3.77e-2) -	5.7036e-1 (1.52e-1) -	4.1779e-2 (4.70e-2) -	8.6339e-1 (1.07e-2) -	8.7086e-1 (2.57e-4)
ZDT6	2	4.2470e-1 (2.45e-3) -	4.3277e-1 (7.94e-4) ≈	3.9550e-1 (1.16e-2) -	4.3342e-1 (2.18e-4) +	4.3003e-1 (1.69e-3) -	4.3119e-1 (9.93e-4) -	4.3297e-1 (4.24e-4)
DTLZ1	3	1.3947e-1 (4.29e-4) -	1.3965e-1 (3.23e-4) -	1.3953e-1 (8.34e-4) -	1.2506e-1 (2.44e-2) -	4.9144e-2 (5.26e-2) -	1.3676e-1 (8.88e-4) -	1.4000e-1 (1.80e-5)
DTLZ2	3	7.4465e-1 (4.54e-5) +	7.4463e-1 (9.42e-5) +	7.4446e-1 (2.90e-4) +	7.0005e-1 (1.84e-3) -	7.3781e-1 (1.87e-3) -	7.2068e-1 (4.67e-3) -	7.4425e-1 (2.98e-4)
DTLZ3	3	3.2185e-1 (3.07e-1) -	3.6040e-1 (2.91e-1) -	2.3777e-1 (2.55e-1) -	3.3787e-1 (3.03e-1) -	0.0000e+0 (0.00e+0) -	5.1204e-1 (2.44e-1) -	6.1475e-1 (2.71e-1)
DTLZ4	3	5.3426e-1 (2.31e-1) ≈	6.7613e-1 (1.26e-1) -	7.4451e-1 (1.18e-4) +	6.7798e-1 (3.63e-2) -	5.3580e-1 (8.39e-2) -	6.6337e-1 (1.14e-1) -	7.4378e-1 (4.70e-4)
DTLZ5	3	1.2107e-1 (2.07e-5) -	1.2860e-1 (8.25e-4) +	9.9884e-2 (5.94e-3) -	1.2934e-1 (5.40e-5) +	1.3247e-1 (1.62e-4) +	1.3228e-1 (6.28e-4) +	1.2675e-1 (2.11e-5)
+/-/≈		1/8/1	2/7/1	2/8/0	2/8/0	1/9/0	1/9/0	

本文绘制出求解 ZDT 和 DTLZ 问题 Pareto 前沿分布图来直观的了解改算法的性能。图 2 展示了 MODE/DMSM 算法在 ZDT、DTLZ 系列测试函数上某次独立运行的 Pareto 前沿分布图，从图 2(a)~(j)可以看出 MODE/DMSM 在 ZDT、DTLZ 系列测试函数上能获得所得到的非支配解集能够较好的逼近真实的 PF 并且分布均匀。

终上所述，针对 ZDT 和 DTLZ 测试函数上，MODE/DMSM 算法无论从 GD、IGD 还是 HV 标准下都比 MOEAD、NSGAIII、RVEA、MOEADDE、NSLS、MOEAIGDNS 的收敛性和多样性要好。MODE/DMSM 算法在处理多目标问题有较好的性能，进一步体现了算法的有效性。

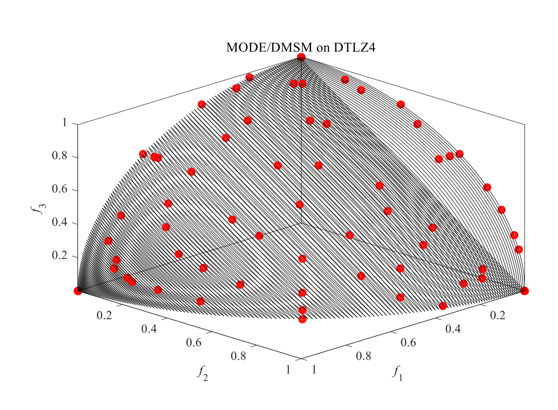


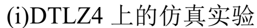
(a) ZDT1 上的仿真实验　　(b)ZDT2 上的仿真实验

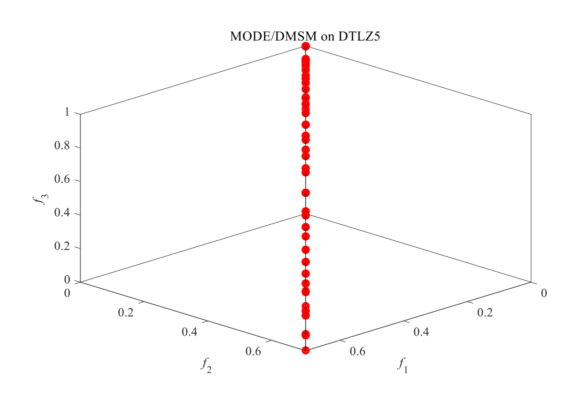
(c) ZDT3 上的仿真实验　　(d)ZDT4 上的仿真实验

(e) ZDT6 上的仿真实验　　(f) DTLZ1 上的仿真实验

(g)DTLZ2 上的仿真实验　　(h)DTLZ3 上的仿真实验




(i)DTLZ4 上的仿真实验



(j)DTLZ5 上的仿真实验

图 2 MODE/DMSM 算法在 ZDT、DTLZ 函数上的 Pareto 前沿分布图

4　结束语

针对多目标问题，本文提出了一种基于分解和多策略变异的多目标差分进化算法 MODE/DMSM。本文运用一种改进 Tchebycheff 分解方法将一个 MOP 分解为一组单目标优化子问题，通过标准边界交叉（NBI）技术生成一系列均匀分布的权重向量，并采用了高效非支配排序有效保存了非支配解和一部分较优的解，达到增加收敛性同时增加其多样性，同时，所提出多策略在进化前期有效保持了种群的多样性，后期达到了收敛性和多样性的平衡。MODE/DMSM 在多目标测试函数集 ZDT 和 DTLZ 的 10 个测试函数上进行了仿真实验，并与 MOEAD、NSGAIII、RVEA、MOEADDE、NSLS 和 MOEAIGDNS 算法进行了对比，结果表明所提出的算法在求解两个或三个目标的多目标问题时在收敛性和多样性具有一定的优势。

致谢：本文的实验环境是安徽大学 BIMK 团队开发的多目标进化算法 PlatEMO 开源平台。对 BIMK 团队提供的帮助，在此致以衷心的感谢！

参考文献：

[1] . Zhou A, Qu B Y, Li H, *et al.* Multiobjective evolutionary algorithms: a survey of the state of the art [J]. Swarm and Evolutionary Computation, 2011, 1 (1): 32-49.

[2] . Deb K, Pratap A, Agarwal S, *et al.* A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. IEEE Trans on Evolutionary Computation, 2002, 6 (2): 182-97.

[3] . Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm [R]. Zurich: Swiss Federal Institute of Technology, 2001.

[4] . Zitzler E, Künzli S. Indicator-based selection in multiobjective search [C]// Proc of International Conference on Parallel Problem Solving from Nature. [S. l.] : Springer, 2004.

[5] . Bader J, Zitzler E. HypE: an algorithm for fast hypervolume-based many-objective optimization [J]. Evolutionary Computation, 2011, 19 (1): 45-76.

[6] . Zhang Q, Li H. MOEA//D: a multiobjective evolutionary algorithm based on decomposition [J]. IEEE Trans on Evolutionary Computation, 2007, 11 (6): 712-731.

[7] . Das S, Suganthan P N. Differential evolution: a survey of the state-of-the-art [J]. IEEE Trans on Evolutionary Computation, 2011, 15 (1): 4-31.

[8] . Zhong J H, Zhang J. Adaptive multi-objective differential evolution with stochastic coding strategy [C]// Proc of the 13th Annual Conference on Genetic and Evolutionary Computation. New York: ACM Press, 2011: 665-72.

[9] . Venske S M, Gonçalves R A, Delgado M R. ADEMO//D: multiobjective optimization by an adaptive differential evolution algorithm [J]. Neurocomputing, 2014, 127: 65-77.

[10] . Jiang S, Yang S. An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts [J]. IEEE Trans on Cybernetics. 2016, 46 (2): 421-437.

[11] . Li H, Zhang Q. Multiobjective optimization problems with complicated Pareto sets, MOEA//D and NSGA-II [J]. IEEE Trans on Evolutionary Computation, 2009, 13 (2): 284-302.

[12] . 周爱民，张青富，张桂戌. 一种基于混合高斯模型的多目标进化算法 [J]. 软件学报, 2014, 25 (5): 913-28.

[13] . Das S, Mullick S S, Suganthan P N. Recent advances in differential evolution – An updated survey [J]. Swarm and Evolutionary Computation, 2016, 27: 1-30.

[14] . Yuan Y, Xu H, Wang B, *et al.* Balancing Convergence and diversity in decomposition-based many-objective optimizers [J]. IEEE Trans on Evolutionary Computation, 2016, 20 (2): 180-198.

[15] . Zhao S Z, Suganthan P N, Zhang Q. Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes [J]. IEEE Trans on Evolutionary Computation, 2012, 16 (3): 442-446.

[16] . Das I, Dennis J E. Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems [J]. SIAM Journal on Optimization, 1998, 8 (3): 631-657.

[17] . Zhang X, Tian Y, Cheng R, *et al.* An efficient approach to nondominated sorting for evolutionary multiobjective optimization [J]. IEEE Trans on Evolutionary Computation, 2015, 19 (2): 201-13.

[18] . 王亚辉，吴金妹，贾晨辉. 基于动态种群多策略差分进化模型的多目标进化算法 [J]. 电子学报, 2016, 44 (6): 1472-80.

[19] . Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints [J]. IEEE Trans on Evolutionary Computation, 2014, 18 (4): 577-601.

[20] . Cheng R, Jin Y, Olhofer M, *et al.* A reference vector guided evolutionary algorithm for many-objective optimization [J]. IEEE Trans on Evolutionary Computation, 2016, 20 (5): 773-91.

[21] . Chen B, Zeng W, Lin Y, *et al.* A new local search-based multiobjective optimization algorithm [J]. IEEE Trans on Evolutionary Computation, 2015, 19 (1): 50-73.

[22] . Tian Y, Zhang X, Cheng R, *et al.* A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric [C]// Proc of IEEE Congress on Evolutionary Computation. 2016.

[23] . Tian Y, Cheng R, Zhang X, *et al.* PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [Educational Forum] [J]. IEEE Computational Intelligence Magazine, 2017, 12 (4): 73-87.

[24] . Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results [J]. Evolutionary Computation, 2000, 8 (2): 173-195.

[25] . Deb K, Thiele L, Laumanns M, *et al.* Scalable test problems for evolutionary multiobjective optimization [C]// Evolutionary Multiobjective Optimization Theoretical Advances and Applications. 2005: 105-45.

[26] . Veldhuizen D A V, Lamont G B. On measuring multiobjective evolutionary algorithm performance [C]// Proc of Congress on Evolutionary Computation. 2000.

[27] . Zhang Q, Zhou A, Jin Y. RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm [J]. IEEE Trans on Evolutionary Computation, 2008, 12 (1): 41-63.

[28] . Yuan Y, Xu H, Wang B, *et al.* A new dominance relation-based evolutionary algorithm for many-objective optimization [J]. IEEE Trans on Evolutionary Computation, 2016, 20 (1): 16-37.